(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 May 2002 (16.05.2002)

PCT

(10) International Publication Number
WO 02/39699 A1

(51) International Patent Classification⁷: H04L 29/06, 29/12

(21) International Application Number: PCT/US00/42125

(22) International Filing Date:
9 November 2000 (09.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (*for all designated States except US*): ENTERA, INC. [US/US]; 40971 Encyclopedia Circle, Fremont, CA 94538 (US).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): SCHARBER, John, M. [US/US]; 1772 Beachwood Way, Pleasanton, CA 94566 (US).

(74) Agents: FAHMI, Tarek, N. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

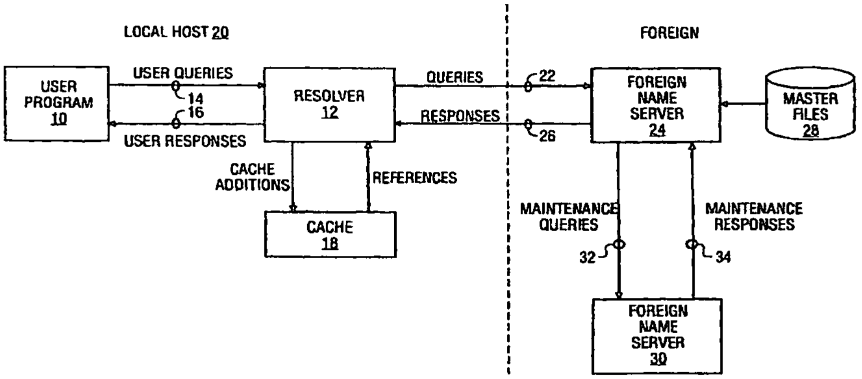
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: DOMAIN NAME SYSTEM EXTENSIONS TO SUPPORT REVERSE PROXY OPERATIONS AND LAYER-7 REDIRECTION



(57) Abstract: A domain name system (DNS) for a computer network (e.g., the Internet) includes a first record resource (RR) type that identifies one or more reverse proxy servers for a specified host within the network according to its network address, for example its Internet Protocol (IP) address and/or its fully qualified domain name. The DNS may also include a second RR type that identifies origin servers from which content may be requested. Preferably, the second RR type includes a weight factor field, the weight factors defining an order in which the hosts should be contacted and/or load-balancing characteristics for the hosts. In some cases, the weight factor for each host may be dynamically generated, for example based on geographic location. Using these RR types, a requesting client may be directed to the one of a number of reverse proxy servers that is geographically closest to the client, as determined using the requesting client's IP address. To facilitate this operation, the requesting client's IP address is used to direct a location query request to the requesting client and geographic location information from the reverse proxy servers are gathered, prior to directing the requesting client to the closest of the reverse proxy servers.

WO 02/39699 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

DOMAIN NAME SYSTEM EXTENSIONS TO SUPPORT REVERSE PROXY OPERATIONS AND LAYER-7 REDIRECTION

FIELD OF THE INVENTION

The present invention relates to a scheme for supporting the use of reverse proxies and application-layer redirection within computer networks, such as the Internet, that utilize a domain name system for identifying network resources.

BACKGROUND

The Internet is a vast and expanding network of networks of computers and other devices linked together by various telecommunications media, enabling all these computers and other devices to exchange and share data. Sites on the Internet provide information about a myriad of corporations and products, as well as educational, research and entertainment information and services. Many millions of people worldwide use the Internet today, with even larger numbers predicted to be on the "net" in a matter of years.

A computer or resource that is attached to the Internet is often referred to as a "host." Examples of such resources include conventional computer systems that are made up of one or more processors, associated memory (typically volatile and non-volatile) and other storage devices and peripherals that allow for connection to the Internet or other networks (e.g., modems, network interfaces and the like). In most cases, the hosting resource may be embodied as hardware and/or software components of a server or other computer system that includes an interface, which allows for some dialog with users thereof. Generally, such a server will be accessed through the Internet (e.g., via Web browsers such as Netscape's Navigator™ and Communicator™ and Microsoft's Internet Explorer™) in the conventional fashion.

Briefly, if an Internet user desires to establish a connection with a host (e.g., to view a Web page located thereat), the user might enter into a Web browser program the URL (or Web address) corresponding to that host. One example of such a URL is "http://www.company.com". In this example, the first element of the URL is a transfer protocol (most commonly, "http" standing for hypertext transfer protocol, but others include "mailto" for electronic mail, "ftp" for file transfer protocol, and "nntp" for network news transfer protocol). The remaining elements of this URL (in this case, "www" standing for World Wide Web--the Internet's graphical user interface--and "company.com") are an alias for the "fully qualified domain name" of the host.

Each fully qualified domain name, in its most generic form, includes three elements. Taking "computer.host.com" as an example, the three elements are the hostname ("computer"), a domain name ("host") and a top-level domain ("com"). Further, each fully qualified domain name is unique throughout the Internet and corresponds to a numerical Internet Protocol (IP) address. IP addresses facilitate communications between hosts and clients in the same way that physical addresses (e.g., 123 Main Street, Anytown, Anycity) facilitate correspondence by regular mail. Each IP address is made up of four groups of numbers separated by decimals. Thus, in the case of the hypothetical host "computer.domain.com", the corresponding IP address might be 123.456.78.91. A given host looks up the IP addresses of other hosts on the Internet through a system known as the domain name service (DNS). It should be recognized that DNS is a logical mapping and may not return a one-to-one association between hosts and addresses.

Specific details of DNS can be found in P. Mockapetris, RFC 1034: "Domain Names - Concepts and Facilities," November 1987, and P. Mockapetris, RFC 1035: "Domain Implementation and Specification," November 1987, both available from the Internet Engineering Task Force at www.ietf.org. Briefly though, the goal of DNS is to provide a mechanism for naming resources in such a way that the names are usable in and by different hosts, networks, protocol families, internets, and administrative organizations. From a user's point of view, domain names may be passed as arguments in a query to a local agent, called a resolver, which then retrieves information (e.g., IP addresses) associated with the domain name.

The database that makes up the domain space (i.e., the database of domain names) is distributed among various name servers (so-called DNS servers) throughout the Internet. When a resolver processes a user query it asks a known name server for the requested information; in return, the resolver either receives the desired information or a referral to another name server. Using these referrals, resolvers learn the identities and contents of other name servers.

Name servers manage two kinds of data: authoritative and non-authoritative that may be cached. Answers from authoritative serves are considered trusted. Authoritative data is held in sets called zones; each zone is the complete database for a particular "pruned" sub-tree of the domain space. Each name server periodically checks to make sure that its zones are up to date, and if not, obtains a new copy of updated zones from master files stored locally or in another name server. Cached data is data acquired by a local resolver. This type data may be incomplete, but improves the performance of the retrieval process when non-local data is repeatedly accessed. Cached data is eventually discarded by a timeout mechanism.

A host can participate in the domain name system in a number of ways, depending on whether the host runs programs that retrieve information from the domain system, name servers that answer queries from other hosts, or various combinations of both functions. One common configuration is shown in **Figure 1**. User programs 10 interact with the domain name space through resolvers 12; the format of user queries 14 and user responses 16 is specific to the host and its operating system. The resolver 12 and its cache 18 may be part of the local host 20 operating system. Less capable hosts may choose to implement a resolver as a subroutine to be linked in with every program that needs its services. Indeed, this seems to be the most common implementation today. Resolvers 12 answer user queries 14 with information they acquire via queries 22 to foreign name servers 24 and the local cache 18. Note that the resolver 12 may have to make several queries to several different foreign name servers to answer a particular user query 14, and hence the resolution of a user query may involve several network accesses and an arbitrary amount of time. The queries 22 to foreign name servers 24 and the corresponding responses 26 have a standard format described in the above-cited RFCs.

Depending on its capabilities, a name server 24 could be a stand alone program on a dedicated computer system or a process or processes on a large timeshared host. As shown, a simple configuration might allow for a name server 24 to acquire information about one or more zones by reading master files 28 from a local file system, and answers queries 22 about those zones that arrive from resolvers 12.

The DNS zones can be made redundant by having more than one name server. Designated secondary servers can acquire zones and check for updates from the primary server using the zone transfer protocol of the DNS. Thus, the name server 24 periodically establishes a virtual circuit to another name server 30 to acquire a copy of a zone or to check that an existing copy has not changed. The maintenance messages 32 and 34 sent and received for these activities follow the same form as queries and responses, but the message sequences are somewhat different. It should be recognized that the system depicted in **Figure 1** is not meant to be fully descriptive of the Internet's DNS, rather it is used merely to identify relevant components thereof.

As the Internet has grown, the use of reverse proxies, combined with so-called Layer-7 redirection, has emerged as one of the primary tools for scaling large Internet sites. "Reverse proxy" is the name for certain alternate uses of a proxy server. For example, proxies can be used outside a firewall to represent a secure content server to outside clients, preventing direct, unmonitored access to that server's data. Proxies can also be used for replication (i.e., caching); that is, multiple proxies can be attached in front of a heavily used server for load balancing or distributed at various geographical sites to mirror content available at an origin server. The goal of such geographic distribution is to ease the burden on an origin server and reduce network congestion due to multiple client requests for similar content. Thus, the proxy servers act as go-betweens for client requests to the real server. The proxy servers cache the requested documents and, if there is more than one proxy server, DNS can route the requests among the proxies using, for example, a "round-robin" selection of their IP addresses. The various clients use the same URL each time, but the route the request takes might go through a different proxy each time.

More recently, others have attempted to apply a proxy redirection strategy that is somewhat more sophisticated than simple round-robin distribution. So-called Layer-7 (referring to the application layer of the open systems interconnect model for networks) redirection is an attempt to divert client requests to proxies based on application-layer criteria. One example is the attempt to direct requests to that proxy which is geographically closest to the requesting client. Although some ability to perform such redirections has been achieved, redirection based on true geographic location has not yet been accomplished.

To understand why this is so, consider the nature of the problem. When a client makes a request for a URL, that request is directed to a name server that tries to resolve the URL into an IP address associated with the host. If the name server is unable to resolve the address, it is forced to ask another name server higher in the domain name hierarchy for the information. This process repeats, until the name server that is able to resolve the address (the so-called authoritative server) is located. At this point, the original name server should provide not the address of the origin server (i.e., the true source of the requested content), but rather the IP address of the closest proxy server to the requesting client. However, the identity of the client is lost. Under the current DNS protocol because the only information provided to the name server is the IP address of the last DNS resolver from which the request was transmitted. Using this information along with route table information, a DNS server can attempt to determine the shortest path distance between an RPS and the last resolving server, however, identity of that client is hidden from the name server; the only information that may be provided is some network topology information identifying a route that the client request took to get to the name server. The RPS may or may not be geographically close to the requesting client, so any attempt to redirect the client to a geographically close proxy can only be as good as the underlying assumption that the name server is close to the client -- an unknowable proposition.

Other schemes that attempt to provide a requesting client with the address of the "closest" proxy server involve transmitting messages from the name server to the identified proxy servers and measuring the response time from those proxies. The proxy with the fastest response time is assumed to be the closest and the client request is resolved to the address of that proxy. This method again assumes that the client is geographically close to the name server that is making the time measurements, and will not ensure that the client is always directed to the closest proxy. Thus, what is needed is a scheme for providing redirection to proxies that avoids the shortcomings of these and other prior schemes.

SUMMARY OF THE INVENTION

In one embodiment, a domain name system (DNS) for a computer network (e.g., the Internet) that includes a record resource (RR) type that identifies one or more reverse proxy servers for a specified host within the network. In some cases, the host may be specified according to a network address, for example an Internet Protocol (IP) address. In other cases, the host may be specified according to its fully qualified domain name. In general, the reverse proxy servers are also specified according to their fully qualified domain name, or other identifying characteristics. The DNS may also include a second RR type that identifies one or more origin servers from which content may be requested.

In a further embodiment, one or more reverse proxy servers for a network host are identified through a domain name system resource record (RR) that may be returned in response to a name query. Preferably, the reverse proxy servers are identified by their

Internet Protocol (IP) addresses. In many cases, the host for which the reverse proxy servers act may be identified according to its fully qualified domain name.

Another embodiment provides a domain name system (DNS) resource record (RR) that includes entries for a network host and its reverse proxy servers. The host is specified in terms of its fully qualified domain name or Internet Protocol address and the reverse proxy servers are specified in terms of their Internet Protocol addresses.

Still another embodiment provides a domain name system (DNS) resource record (RR) that includes entries for a network hosts configured to act a source for specified content. As above, the hosts may be specified in terms of their fully qualified domain names or Internet Protocol addresses. Preferably, the RR includes a weight factor for each host, the weight factors defining an order in which the hosts should be contacted and/or load-balancing characteristics for the hosts. In some cases, the weight factor for each host may be dynamically generated, for example based on geographic location.

In a further embodiment, a domain name query that includes a requesting client Internet Protocol (IP) address as part of a data field is transmitted. The requesting client may then be directed to one of a number of reverse proxy servers according to the client's geographic proximity to that reverse proxy server, as determined using the requesting client's IP address. To facilitate this operation, the requesting client's IP address is used to direct a location query request to the requesting client and geographic location information from the reverse proxy servers are gathered, prior to directing the requesting client to the closest of the reverse proxy servers.

In still another embodiment, a client request may be redirected to one of a number of available reverse proxy servers by determining geographic location information for the client making the request through the use of an Internet Protocol (IP) address of the client transmitted as part of a name query by the client and subsequently directing the client to the physically closest of the available reverse proxy servers. The available reverse proxy servers are preferably identified through a name service resource record type. Prior to directing the client to the physically closest of the available reverse proxy servers, the geographic location of the client is determined by transmitting a location query using the client's IP address.

Other features and advantages of the present invention will be apparent from the following discussion.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements and in which:

Figure 1 illustrates an example of a network that includes a name server configured to operate according to the domain name system.

DETAILED DESCRIPTION

Disclosed herein is a scheme for extending a domain name service to support reverse proxy operation and Layer-7 redirection. In essence, the present scheme defines two new resource record (RR) types for use by DNS. One RR, "RPS", defines a list of IP addresses that should act as reverse proxy servers for a specified IP address. The second RR, "OCS", may be used to define a list of origin content servers from which content can be retrieved. Through the use of these new RRs, proxy servers are provided with the ability to build a reverse proxy map to dynamically determine whether they should act as a reverse proxy for an incoming client request, and, if so, where to retrieve the requested content from. The map may also include static relationship information defined outside of the DNS.

Although discussed with reference to certain illustrated embodiments, upon review of this specification, those of ordinary skill in the art will recognize that the present scheme may find application in a variety of systems, perhaps with one or more minor variations. For example, although discussed primarily with respect to the Internet domain name service, the same methodologies may be applied to other networks where hierarchical domain names are used. Therefore, in the following description the illustrated embodiments should be regarded as exemplary only and should not be deemed to be limiting in scope. Further, it should be kept in mind that some portions of the detailed description that follows are presented in terms of algorithms and symbolic representations (e.g., through the use of flow diagrams, etc.) of operations on data within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the computer science arts to most effectively convey the substance of their work to others skilled in the art.

An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Moreover, unless specifically stated otherwise, it will be appreciated that throughout the description of the present scheme, use of terms such as "processing", "computing, "calculating", "determining", "displaying", "rendering" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices. Again, these are the terms and descriptions commonly used by and among practitioners of ordinary skill in the relevant arts.

As described above, the present scheme provides new RRs to the DNS. To understand the present invention then, it is necessary to first understand how RRs are used in the existing DNS. One use of the RRs within the DNS involves the creation of tree structures for organizing the DNS name space. Thus, RRs are specifications for a tree structured (i.e., hierarchical) name space and data associated with the names. Conceptually, each node and leaf of the domain name space tree names a set of information, and query operations are attempts to extract specific types of information from a particular set. A query names the domain name (i.e., node) of interest and describes the type of resource information that is desired. For example, queries for address resources return IP host addresses.

Every RR has the following attributes: (i) owner (the domain name where the RR is found; often this is implicit); (ii) type (an encoded 16-bit value that specifies the type of the resource in the subject resource record, for example, "A" might identify a host address, "NS" might identify the authoritative name server for the domain, etc.); (iii) class (an encoded 16-bit value which identifies a protocol family or instance of a protocol, for example "IN" might identify the Internet system); (iv) TTL (the time to live of the RR, which describes how long an RR can be cached before it should be discarded); and (v) RDATA (the type and sometimes class dependent data that describes the resource, e.g., for the IN class, a 32-bit IP address). The type, class and TTL fields of a RR are fixed portions of a header, while the RDATA field is a variable length field that fits the needs of the resource being described. The data in the RDATA section of RRs is carried as a combination of binary strings and domain names. The domain names are frequently used as "pointers" to other data in the DNS.

RRs are typically represented in binary form in the packets of the DNS protocol, and are usually represented in highly encoded form when stored in a name server or resolver. In this format, most RRs are shown on a single line, although continuation lines are possible using parentheses. The start of the line gives the owner of the RR. If a line begins with a blank, then the owner is assumed to be the same as that of the previous RR. Following the owner are listed the TTL, type, and class of the RR. The resource data or RDATA section of the RR is given using knowledge of the typical representation for the data. Address RRs use a standard IP address format to contain a 32-bit Internet address.

With this background, the presently proposed RR types can now be introduced. As indicated above, the RR "RPS" defines a list of IP addresses that should act as reverse proxy servers for a specified IP address. The format of the RPS type is:

<hostname> <TTL> <CLASS> RPS <IP>

In this data structure, "hostname" is the IP address or fully qualified domain name that is configured to use a reverse proxy, "TTL" and "CLASS" are used as above, and "IP" is the IP address(es) of a reverse proxy. Thus, for an Internet-based resource type, CLASS will be IN (following the standard DNS protocol) and to define a group of five servers to act as reverse proxies for the host www.computer.com one might create a configuration that looks like the following:

```
www.computer.com     IN  RPS  10.0.0.1.
                              10.0.1.1.
                              10.0.2.1.
                              10.0.3.1.
                              10.0.4.1.
```

Consider then the following scenario. A client makes a request for the host www.computer.com. This request is received at a server capable of providing reverse proxy services (e.g., it has an available cache to store requested content) and having an associated name server. When a request to resolve the host name to an IP address is made to the name server, the response should include the RPS RR above.

The use of this RPS RDATA format allows the proxy server to dynamically determine whether it should act as a reverse proxy for the specified host. Because this is a request from an RPS, as an optimization the DNS server may return a lost of one RPs; that of the RPS performing the query. In general, if the proxy finds its own IP address in the list returned by the name server, it will then add the mapping to its internal mapping table (generally stored in memory) until the TTL reported in the RPS RR has expired. During this time, any client requests for www.computer.com that are received at the server are intercepted and the requested content is played out of the server, acting as a reverse proxy for the origin server (www.computer.com). Because the reverse proxy server only adds entries to its internal list as traffic for a site is generated, the internal list can be significantly smaller than the total list of sites that reverse proxy services are available for. In general, the improved efficiency of list processing should more than compensate for having to perform additional DNS queries.

In one embodiment of the present scheme, the RPS RDATA format is combined with the WKS (well known server) data format to determine what protocols can be reverse proxied on a given server. For example, the WKS data may be appended to the original RPS request to specify the type of content being requested, and the list of servers that should act as reverse proxies may be modified based on this information.

The second new RR type is the OCS (origin content server) RDATA format. This format is used to define a list of servers from which the actual content can be retrieved. It also provides support for load balancing and fail-over operations. The format of the OCS type is:

<hostname> <TTL> <CLASS> OCS <WEIGHT> <IP>

where: "hostname" is the IP address or fully qualified domain name that is configured to use a reverse proxy and must be the same as that used in the RPS configuration; "TTL" and "CLASS" have the meanings discussed above; "WEIGHT" is a field that defines the order in which servers in the list should be tried (if two or more servers in the list have the same WEIGHT, then a client should load balance between them, e.g., using a round-robin load balancing procedure); and "IP" is a field listing the available servers from which the content can be retrieved. In some cases, the WEIGHT parameter may be dynamically generated by a name server, for example based on geographic location.

Thus, an example of an OCS entry with a single primary and secondary server may resemble:

```
www.computer.com  IN  OCS  10  10.1.0.1.
                           20  10.1.1.1.
```

An example of an entry for a pair of primary servers performing load balancing and a single secondary server is:

```
www.computer.com  IN  OCS  10  10.1.0.1.
                           10  10.1.1.1.
                           20  10.1.2.1.
```

The RPS and OCS attributes are used together to build a reverse proxy map. If the server has an entry in the RPS table, then the server should send an OCS query to discover which origin server can accommodate the request.

The WEIGHT factor may be any value in a range 0 - 255, although the use of 0 may be reserved for special functions. Within this range, in one embodiment a WEIGHT of 1 is given the highest priority and 255 the lowest priority, with intervening WEIGHT factors having appropriate priorities according to this scale. Given this range, it is possible for a client to adjust the WEIGHT factors of given servers based on actual response time, error statistics, through put, and the like. Many different schemes for gathering such data can be used and the specific implementation is not critical of the present invention.

The OCS RDATA format can be combined with the WKS data format to determine what protocols can be reverse proxied on a given server. Servers performing Layer-7 redirection may compare the WKS information from both the OCS and RPS to ensure that client requests are redirected to a proxy capable of accommodating all of the expected content types (specified by the WKS data). In one embodiment, the WKS data is appended to the original OCS request to improve performance. If there is more than one RPS, then the WKS record may follow its associated RPS, for example:

RPS | WKS | RPS | WKS

If an OCS record is specified for a given resource, then the PRS should support all the requested WKS(s) specified to act as a reverse proxy server. Likewise, a name server directing traffic to an RPS should not redirect traffic to an RPS that either does not contain a WKS record or cannot accommodate all the required WKS data types of the OCS.

To further provide for true geographic-based redirection, the present scheme also contemplates including the IP addresses and/or geographic locations of clients and/or reverse proxy servers within DNS queries. Currently, DNS queries and responses are carried in a standard message format. The message format has a header containing a number of fixed fields which are always present, and four sections which carry query parameters and
RRs. One field in the header is a four-bit field called an opcode which distinguishes between different types of queries. For example, a specific opcode is used to identify a so-called "standard query". The four sections of a DNS query are:

Question Carries the query name and other query parameters.
Answer Carries RRs which directly answer the query.
Authority Carries RRs which describe other authoritative servers. May optionally carry the SOA RR for the authoritative data in the answer section.
Additional Carries RRs which may be helpful in using the RRs in the other
sections.

The content, but not the format, of these sections varies with header opcode.

A standard query specifies a target domain name (QNAME), query type (QTYPE), and query class (QCLASS) and asks for RRs which match. This type of query makes up the vast majority of DNS queries that are used in the Internet today. The QTYPE and QCLASS fields are each 16-bits long, and are a superset of defined types and classes. The QTYPE field may contain:

<any type> matches just that type. (e.g., A, PTR).
AXFR special zone transfer QTYPE.
MAILB matches all mail box related RRs (e.g. MB and MG).
* matches all RR types.

The QCLASS field may contain:

<any class> matches just that class (e.g., IN, CH).
* matches aLL RR classes.

Using the query domain name, QTYPE, and QCLASS, the name server looks for matching RRs. In addition to relevant records, the name server may return RRs that point toward a name server that has the desired information or RRs that are expected to be useful in interpreting the relevant RRs. For example, a name server that doesn't have the requested information may know a name server that does; a name server that returns a domain name in a relevant RR may also return the RR that binds that domain name to an address.

Because currently name servers are not provided with any information regarding the true geographic location of a requesting client, several assumptions have to be made when considering where to direct the client request (i.e., when attempting to direct the request to the closest reverse proxy server). For example, the name server responding to the request may need to assume that it is geographically co-located with the client. Alternatively, or in addition, geographic proximity may be approximated using a combination of network distance and latency. Experience has shown that these assumptions fail in many cases.

To provide for true geographic-based redirection, the present scheme modifies DNS queries to include the originating client's IP address. Either of two approaches may be used to implement this feature. In one approach, a new opcode (e.g., opcode 16) is added to the DNS, for use in a DNS query header. This new opcode will alert a name server that the associated query includes information allowing for true geographic-based redirection in accordance with the present invention. The use of a new opcode also allows for backwards compatibility with current implementations of the DNS in as much as if the name server does not support opcode 16 functionality, it can return an error message to the requesting server, which server can then retransmit the query and a standard query (opcode 0), without the client-identifying information. In an alternative approach, the requesting client IP address can be appended to the end of the QCLASS portion of a standard query and then decoded by the name server. With this second approach, care must be taken to ensure that any un-initialized data in the answer section that follows the query is not erroneously interpreted as a client IP address.

Exploring the first option (the use of a new opcode) further, when an opcode 16 (note, it should be recognized that the use of opcode 16 herein is for convenience only and the present scheme may be implemented using any not otherwise reserved opcode) query is received by an authoritative server for an "A" RDATA record, that server should perform an internal lookup for all RPSs associated with the requested resource. Next, the name server should perform an LOC (location) query on the IP address/subnet of the originating client to obtain its precise geographic coordinates.

The use of LOC queries is described in detail in C. Davis et al., RFC 1876 "A Means for Expressing Location Information in the Domain Name System" January 1996 (incorporated herein by reference), which describes a mechanism to allow the DNS to carry location information about hosts, networks, and subnets. Under that scheme, the RDATA format of the LOC RR includes fields for the latitude, longitude and altitude of an identified host, network or subnet.

Where the name server successfully retrieves the LOC parameters of the requesting client, it can then retrieve (either using an LOC query or from a cache) the LOC parameters of the available RPSs (identified using the RPS records) and determine the true geographically closest RPS to the requesting client. It should be noted that there are many possible methods for optimizing the internal structure and retrieval of this information and the details of such a search are not critical to the present invention.

One use of the OCS WEIGHT field may be to dynamically determine the weight based on geographic considerations. For example, if there was a global content provider that had already replicated its main content server at several locations throughout the world, it would be desirable to return the OCSs back in an order that represented geographic proximity to a requestor. For example, if OCS servers A, B and C are available in Europe, North America and Asia, respectively, and each OCS has three associated RPSs, it would be desirable to return OCS A for all PRSs in Europe.

In such a situation, load balancing using the OCS WEIGHT field takes on a new meaning. Now, because the actual metric will be dynamically generated based on locality, servers that have the same WEIGHT will be considered geographically co-located. That is, the WEIGHT values that appear in the configuration file serve merely to act as a means of specifying redundant servers for load balancing. To illustrate, consider a situation where www.computer.com is a host associated with three primary OCSs (servers A, B and C), each geographically located in a different region (e.g., server A may be located in Europe, server B in North America and server C in the Asia-pacific region). In addition to the primary servers, three backup servers (A(b), B(b), and C(b)) may be deployed in the same regions (but not necessarily the same physical locations), to provide load-balancing capability. Further, each primary/backup server combination may be associated with three PRSs, each. Thus, the resulting configuration file may resemble the following:

```
www.computer.com        IN      OCS   10      10.1.0.1.
```

10	10.1.1.1.
20	10.1.3.1.
20	10.1.4.1.
30	10.1.5.1.
30	10.1.6.1.

In this example, the weight 10 servers would represent the load-balancing servers for Europe, the weight 20 servers for North America and the weight 30 servers for Asia-Pacific. One could also define weight ranges for different geographic locations,e.g.,

10 – 50	= N. America
51 – 100	= S. America

would require a predetemined scheme for such geographic coding agreed upon between the client and the server. Such schemes could be indicated using the WKS entries described above.

Thus a scheme for extending a domain name service to support reverse proxy operation and Layer-7 redirection has been described. Although the foregoing description and accompanying figures discuss and illustrate specific embodiments, it should be appreciated that the present invention is to be measured only in terms of the claims that follow.

CLAIMS

What is claimed is:

1. A domain name system (DNS) for a computer network comprising a first record resource (RR) type that identifies one or more reverse proxy servers for a specified host within the network.

2. The DNS of claim 1 wherein the network comprises the Internet.

3. The DNS of claim 1 wherein the host is specified according to a network address.

4. The DNS of claim 3 wherein the network address comprises and Internet Protocol (IP) address.

5. The DNS of claim 1 wherein the host is specified according to its fully qualified domain name.

6. The DNS of claim 1 further comprising a second RR type that identifies one or more origin servers from which content may be requested..

7. The DNS of claim 1 wherein the first RR type is specified in a form that identifies the host in terms of its fully qualified domain name or Internet Protocol (IP) address.

8. The DNS of claim 7 wherein the first RR type is further specified in a form that identifies the one or more reverse proxy servers by their associated IP addresses and/or fully qualified domain names.

9. A method, comprising identifying one or more reverse proxy servers for a network host through a domain name system resource record (RR).

10. The method of claim 9 wherein the RR is returned in response to a name query.

11. The method of claim 9 wherein the reverse proxy servers are identified by their Internet Protocol (IP) addresses.

12. The method of claim11 further comprising identifying the host for which the reverse proxy servers act according to its fully qualified domain name.

13. A domain name system (DNS) resource record (RR) comprising entries for a network host and its reverse proxy servers.

14. The DNS of claim 13 wherein the host is specified in terms of its fully qualified domain name or Internet Protocol address.

15. The DNS of claim 13 wherein the reverse proxy servers are specified in terms of their Internet Protocol addresses.

16. A domain name system (DNS) resource record (RR) comprising entries for a network hosts configured to act a source for specified content.

17. The DNS of claim 16 wherein the hosts are specified in terms of their fully qualified domain names or Internet Protocol addresses.

18. The DNS of claim 16 wherein the RR comprises a weight factor for each host, the weight factors defining an order in which the hosts should be contacted.

19. The DNS of claim 16 wherein the RR comprises a weight factor for each host, the weight factors defining load-balancing characteristics for the hosts.

20. The DNS of claim 16 wherein the RR comprises a dynamically generated weight factor for each host.

21. The DNS of claim 16 wherein the RR comprises a geographic mapping indicated by a WKS entry or other defining criteria agreed upon between clients and the hosts.

22. The DNS of claim 20 wherein the weight factors are generated based on geographic location.

23. A method comprising transmitting a domain name query that includes a requesting client Internet Protocol (IP) address as part of a data field.

24. The method of claim 23 further comprising directing the requesting client to one of a number of reverse proxy servers according to the client's geographic proximity to that reverse proxy server as determined using the requesting client's IP address.

25. The method of claim 24 wherein the requesting client's IP address is used to direct a location query request to the requesting client, prior to directing the client to the one of the reverse proxy servers.

26. The method of claim 25 further comprising receiving geographic location information from the reverse proxy servers prior to directing the requesting client to the one of the reverse proxy servers.

27. A method for redirecting a client request to one of a number of available reverse proxy servers comprising determining geographic location information for the client making the request through the use of an Internet Protocol (IP) address of the client

transmitted as part of a name query by the client and subsequently directing the client to the physically closest of the available reverse proxy servers.

28. The method of claim 27 wherein the available reverse proxy servers are identified through a name service resource record type.

29. The method of claim 28 wherein prior to directing the client to the physically closest of the available reverse proxy servers, the geographic location of the client is determined by transmitting a location query to the client using its IP address.

30. A method comprising transmitting geographic coordinates of a requesting client as part of a domain name system (DNS) query initiated by the requesting client.

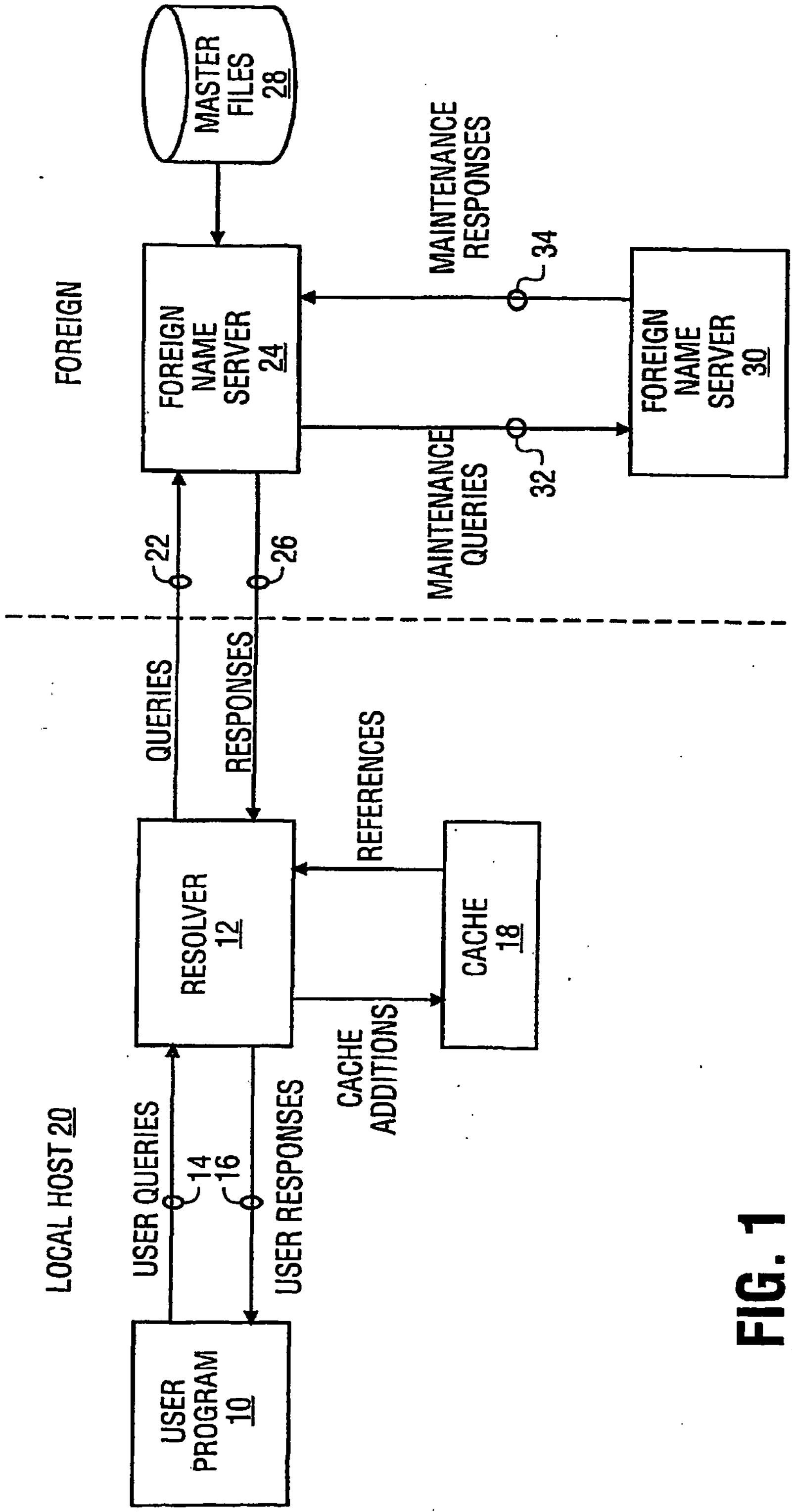

LOCAL HOST 20
USER PROGRAM 10
USER QUERIES
14
16
USER RESPONSES
RESOLVER 12
CACHE ADDITIONS
REFERENCES
CACHE 18
QUERIES
RESPONSES
22
26
FOREIGN
FOREIGN NAME SERVER 24
MASTER FILES 28
MAINTENANCE QUERIES
32
MAINTENANCE RESPONSES
34
FOREIGN NAME SERVER 30

FIG. 1

INTERNATIONAL SEARCH REPORT

Inte[redacted]onal Application No
PCT/US 00/42125

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04L29/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	NOTTINGHAM M: "On defining a role for demand-driven surrogate origin servers" COMPUTER COMMUNICATIONS, ELSEVIER SCIENCE PUBLISHERS BV, AMSTERDAM, NL, vol. 24, no. 2, 1 February 2000 (2000-02-01), pages 215-221, XP004228463 ISSN: 0140-3664	1-5,7-17
Y	abstract	24,27,28
A	page 215, left-hand column, line 1 -page 216, right-hand column, line 32	23

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 29 August 2001	Date of mailing of the international search report 07/09/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Lievens, K

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	M. GREEN, B. CAIN, G. TOMLINSON: "CDN Peering Architectural Overview" INTERNET-DRAFT , ‘Online! 20 October 2000 (2000-10-20), pages 1-34, XP002176084 Retrieved from the Internet: <URL:http://www.alternic.org/drafts/drafts-g-h/draft-green-cdnp-gen-arch-01.txt> ‘retrieved on 2001-08-29!	1-5, 7-17,23
A	abstract paragraph ‘0002! - paragraph ‘04.3! ---	27
X	EP 0 817 444 A (SUN MICROSYSTEMS INC) 7 January 1998 (1998-01-07)	16,17, 23,30
Y	abstract column 1, line 1 -column 6, line 45 figure 3 ---	18,19, 24,27,28
Y	A. GULBRANDSEN, P. VIXIE, L. ESIBOV: "A DNS RR for specifying the location of services (DNS SRV)" REQUEST FOR COMMENTS: 2782 , ‘Online! February 2000 (2000-02), pages 1-8, XP002176085 Retrieved from the Internet: <URL:http://www.faqs.org/rfcs/rfc2782.html> ‘retrieved on 2001-08-29! the whole document ---	18,19
A	P. MOCKAPETRIS: "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION" REQUEST FOR COMMENTS: 1035, ‘Online! November 1987 (1987-11), pages 1-39, XP002176086 Retrieved from the Internet: <URL:http://www.faqs.org/rfcs/rfc1035.html> ‘retrieved on 2001-08-29! cited in the application paragraph ‘03.2! paragraph ‘03.3! -----	1-22,28

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/42125

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0817444 A	07-01-1998	US 6154777 A	28-11-2000
		JP 10126445 A	15-05-1998